Started on	Wednesday, 8 May 2024, 1:16 PM
State	Finished
Completed on	Wednesday, 8 May 2024, 2:40 PM
Time taken	1 hour 24 mins
Grade	**344.00** out of 350.00 (**98**%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Lengkapi header Stack.java berikut. Silahkan kumpulkan **Stack.java** yang telah kalian lengkapi!

Java 8 ⬍

⚙ Stack.java

Score: 100

Blackbox
Score: 100
Verdict: Accepted
Evaluator: Exact

No	Score	Verdict	Description
1	15	Accepted	0.08 sec, 28.40 MB
2	15	Accepted	0.09 sec, 28.96 MB
3	15	Accepted	0.08 sec, 27.84 MB
4	15	Accepted	0.08 sec, 29.99 MB
5	20	Accepted	0.10 sec, 28.32 MB
6	20	Accepted	0.08 sec, 28.25 MB

Time limit	1 s
Memory limit	64 MB

Silahkan lengkapi Deque.java berikut. Setelah itu, manfaatkan Deque yang telah kalian buat untuk mengimplementasikan isPalindrome pada Palindrome.java. Silahkan kumpulkan Deque.java dan Palindrome.java dalam sebuah zip bernama **Deque.zip**

Java 8 ⬍

📄 Deque.zip

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	8	Accepted	0.11 sec, 28.35 MB
2	8	Accepted	0.28 sec, 28.85 MB
3	8	Accepted	0.31 sec, 28.70 MB
4	8	Accepted	0.17 sec, 28.72 MB
5	8	Accepted	0.21 sec, 27.98 MB
6	8	Accepted	0.16 sec, 29.95 MB
7	8	Accepted	0.21 sec, 28.55 MB
8	8	Accepted	0.14 sec, 28.36 MB
9	8	Accepted	0.13 sec, 30.19 MB
10	8	Accepted	0.15 sec, 29.15 MB
11	10	Accepted	0.18 sec, 28.39 MB
12	10	Accepted	0.11 sec, 28.59 MB

Time limit	1 s
Memory limit	64 MB

Kita akan belajar untuk mereplikasi Monadic Error Handling dari functional programming. Dengan monadic error handling, kita dapat merepresentasikan suatu value yang mungkin merupakan error. Dengan menggunakan pattern ini, kita dapat mengelola suatu value yang mungkin error tanpa melalui try-catch dan membuat programmer lebih aware jika value yang dikelola dapat menghasilkan error

Diberikan suatu interface IResult.java sebagai berikut beserta penjelasan dalam bentuk comment. Implementasikan interface tersebut dalam suatu class di dalam file Result.java

Java 8 ◆

⚙ Result.java

Score: 100

Blackbox
Score: 100
Verdict: Accepted
Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.13 sec, 28.52 MB
2	20	Accepted	0.09 sec, 26.14 MB
3	20	Accepted	0.10 sec, 29.76 MB
4	20	Accepted	0.10 sec, 28.18 MB
5	20	Accepted	0.11 sec, 28.38 MB

Question **4**

Partially correct

Mark 44.00 out of 50.00

Time limit	1 s
Memory limit	64 MB

Kita akan belajar functional programming melalui materi Generics dan Java API untuk memanipulasi List Java

Diberikan suatu interface `IStreamer.java` sebagai berikut beserta penjelasan dalam bentuk comment. Tolong implementasikan interface tersebut dalam suatu class di dalam file `Streamer.java`

Note:

- `Function<T,V>` merupakan suatu interface yg disediakan `java.util.function.Function` untuk merepresentasikan lambda function di Java

- Contoh pemakaian

```java
Function<Integer, Double> f = (Integer x) -> x * 2.0;
Double result = f.apply(2); // value: 4.0
```

Java 8 ⬍

⚙ Streamer.java

Score: 44

Blackbox

Score: 44

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	2	Accepted	0.08 sec, 28.04 MB
2	2	Accepted	0.17 sec, 35.73 MB
3	2	Accepted	0.18 sec, 34.09 MB
4	2	Accepted	0.17 sec, 35.24 MB
5	2	Accepted	0.18 sec, 34.20 MB
6	2	Accepted	0.17 sec, 34.00 MB
7	2	Accepted	0.19 sec, 35.40 MB
8	3	Accepted	0.18 sec, 34.53 MB
9	3	Accepted	0.07 sec, 26.23 MB
10	3	Accepted	0.08 sec, 28.98 MB
11	3	Accepted	0.08 sec, 28.79 MB
12	0	Wrong answer	0.09 sec, 28.51 MB
13	3	Accepted	0.09 sec, 28.68 MB

No	Score	Verdict	Description
14	3	Accepted	0.07 sec, 29.02 MB
15	3	Accepted	0.07 sec, 30.39 MB
16	3	Accepted	0.07 sec, 28.54 MB
17	3	Accepted	0.18 sec, 33.54 MB
18	3	Accepted	0.18 sec, 33.98 MB
19	0	Wrong answer	0.07 sec, 29.79 MB

Jump to...

No	Score	Verdict	Description